2.9 Incomplete Elliptic Integrals

A. Purpose

An integral of the form

$$\int R\left(t, P(t)^{1/2}\right) dt \tag{1}$$

in which P(t) is a polynomial of the third or fourth degree that has no multiple roots, and R is a rational function of t and P$(t)^{1/2}$, is either elementary, or is an *elliptic integral*. It is always possible to express integrals of the form of Eq. (1) linearly in terms of elementary functions and three *elliptic integrals of canonical form*. These functions are described more completely in [1] and [2]. Several canonical forms have been proposed, but the most widely used are due to Jacobi, Legendre and Carlson. In each of Eqs. (2)–(4) we present first Jacobi's and then Legendre's form of the canonical elliptic integrals:

$$\begin{aligned} F(\varphi, k) &= \int_0^y \left(1-t^2\right)^{-1/2} \left(1-k^2t^2\right)^{-1/2} dt \\ &= \int_0^\varphi \left(1-k^2\sin^2\theta\right)^{-1/2} d\theta \end{aligned} \tag{2}$$

$$\begin{aligned} E(\varphi, k) &= \int_0^y \left(1-t^2\right)^{-1/2} \left(1-k^2t^2\right)^{1/2} dt \\ &= \int_0^\varphi \left(1-k^2\sin^2\theta\right)^{1/2} d\theta \end{aligned} \tag{3}$$

$$\begin{aligned} &\Pi(\varphi, \alpha, k) \\ &= \int_0^y \left(1-\alpha^2t^2\right)^{-1} \left(1-t^2\right)^{-1/2} \left(1-k^2t^2\right)^{-1/2} dt \\ &= \int_0^\varphi \left(1-\alpha^2\sin^2\theta\right)^{-1} \left(1-k^2\sin^2\theta\right)^{-1/2} d\theta \end{aligned} \tag{4}$$

in which $y = \sin\varphi$. If φ is equal to $\pi/2$, the integrals are said to be *complete*, otherwise they are *incomplete*. Carlson's forms of the canonical elliptic integrals are

$$R_D(a,b,c) = \frac{3}{2}\int_0^\infty (t+a)^{-1/2}(t+b)^{-1/2}(t+c)^{-3/2}\,dt \tag{5}$$

in which a and b are nonnegative such that $a + b > 0$ and c is positive; if either a or b is zero, the integral is *complete*, otherwise it is *incomplete*,

$$R_F(a,b,c) = \frac{1}{2}\int_0^\infty (t+a)^{-1/2}(t+b)^{-1/2}(t+c)^{-1/2}\,dt \tag{6}$$

in which a, b and c are nonnegative and at most one of them is zero; if one of a, b or c is zero, the integral is *complete*, otherwise it is *incomplete*, and

$$\begin{aligned} &R_J(a,b,c,r) \\ &= \frac{3}{2}\int_0^\infty (t+r)^{-1}(t+a)^{-1/2}(t+b)^{-1/2}(t+c)^{-1/2}\,dt \end{aligned} \tag{7}$$

in which a, b and c are nonnegative, and at most one of them is zero, and r is nonzero; if one of a, b or c is zero, the integral is *complete*, otherwise it is *incomplete*. Notice that $R_D(a,\, b,\, c) = R_J(a,\, b,\, c,\, c)$. But the necessity to compute $R_D(a,\, b,\, c)$ arises frequently in practice, and a procedure especially tailored to compute $R_D(a,\, b,\, c)$ is more efficient than computing $R_J(a,\, b,\, c,\, c)$. The function $R_C(a,\, b) = R_F(a,\, b,\, b)$ is elementary, but also appears frequently. A procedure is provided to compute $R_C(a,\, b)$.

Identify a, b and c such that $a \le b \le c$, and assume $a < c$. Then

$$c^{3/2} R_D(a,b,c) = \frac{3}{k^2\sin^3\varphi}\left[F(\varphi,k) - E(\varphi,k)\right] \tag{8}$$

$$c^{1/2} R_F(a,b,c) = \frac{F(\varphi,k)}{\sin\varphi} \tag{9}$$

$$c^{3/2} R_J(a,b,c,r) = \frac{3}{\alpha^2\sin^3\varphi}\left[\Pi(\varphi,\alpha,k) - F(\varphi,k)\right] \tag{10}$$

where $\cos^2\varphi = a/c$, $k^2 = (c-b)/(c-a)$ and $\alpha^2 = (c-r)/(c-a)$.

The subprograms described in this chapter evaluate the canonical forms of incomplete elliptic integrals, using either the Legendre or the Carlson parameterization.

B. Usage

B.1 Program Prototype, Single Precision, Legendre's Form, E and F

REAL PHI, K, F, E

INTEGER IERR

Assign values to PHI and K.

CALL SELEFI (PHI, K, F, E, IERR)

B.1.a Argument Definitions

PHI [in] Argument, φ, of the elliptic integral. Require $|\text{PHI}| \le \pi/2$.

K [in] Modulus, k. Require $|\text{K}| \le 1.0$.

F [out] F(φ, k) with φ given by PHI and k given by K.

E [out] E(φ, k) with φ given by PHI and k given by K.

IERR [out] status indicator:

0 = no errors
1 = Magnitude of argument too large, $|\text{PHI}| > \pi/2$.
2 = Magnitude of Modulus too large, $|\text{K}| > 1.0$.
3 = $|\text{PHI}| = \pi/2$ and $|\text{K}| = 1$, F is infinite.

B.2 Program Prototype, Single Precision, Legendre's Form, Π

REAL PHI, K2, ALPHA2, PI

INTEGER IERR

Assign values to PHI, K2 and ALPHA2.

CALL SELPII (PHI, K2, ALPHA2, PI, IERR)

B.2.a Argument Definitions

PHI [in] Argument, φ, of the elliptic integral. Require $|\text{PHI}| \le \pi/2$.

K2 [in] Square of the modulus, k^2. Require $k^2 \sin^2 \varphi \le 1.0$. See Section E.

ALPHA2 [in] Characteristic, α^2. Require $\alpha^2 \sin^2 \varphi \le 1.0$. See Section E.

PI [out] $\Pi(\varphi, \alpha, k)$, with φ given by PHI, α^2 given by ALPHA2 and k^2 given by K2.

IERR [out] status indicator. If IERR = 0, there were no errors. Other values are produced by procedures SRFVAL and SRJVAL (see Sections B.5 and B.6) which are used in computing $\Pi(\phi, \alpha, k)$.

B.3 Program Prototype, Single Precision, Carlson's Form, R_C

REAL X, Y, RC

INTEGER IERR

Assign values to X and Y.

CALL SRCVAL (X, Y, RC, IERR)

B.3.a Argument Definitions

X, Y [in] Arguments of the elliptic integral. Require X ≥ 0, Y $\ne 0$. See Section E.

RC [out] The computed value of R_C(X, Y).

IERR [out] Status indicator:

0 = no errors
1 = X < 0.0 or Y = 0.0.
2 = X + |Y| too small (See Section E).
3 = X or |Y| or X + |Y| too large (See Section E).
4 = Y < 0 and |Y| too large and X too small (See Section E).

B.4 Program Prototype, Single Precision, Carlson's Form, R_D

REAL X, Y, Z, RD

INTEGER IERR

Assign values to X, Y and Z.

CALL SRDVAL (X, Y, Z, RD, IERR)

B.4.a Argument Definitions

X, Y, Z [in] Arguments of the elliptic integral. Require X ≥ 0, Y ≥ 0, X + Y > 0, Z > 0. See Section E.

RD [out] The computed value of R_D(X, Y, Z).

IERR [out] Status indicator:

0 = no errors
1 = X < 0.0 or Y < 0.0 or Z < 0.0.
2 = X + Y too small or Z too small (See Section E).
3 = X or Y or Z too large (See Section E).

B.5 Program Prototype, Single Precision, Carlson's Form, R_F

REAL X, Y, Z, RF

INTEGER IERR

Assign values to X, Y and Z.

CALL SRFVAL (X, Y, Z, RF, IERR)

B.5.a Argument Definitions

X, Y, Z [in] Arguments of the elliptic integral. Require X ≥ 0, Y ≥ 0, Z ≥ 0, at most one of X, Y or Z equal zero. See Section E.

RF [out] The computed value of R_F(X, Y, Z).

IERR [out] Status indicator:

0 = no errors
1 = X < 0.0 or Y < 0.0 or Z < 0.0.
2 = X + Y or X + Z or Y + Z too small (See Section E).
3 = X or Y or Z too large (See Section E).

B.6 Program Prototype, Single Precision, Carlson's Form, R_J

REAL X, Y, Z, R, RJ

INTEGER IERR

Assign values to X, Y, Z and R.

CALL SRJVAL (X, Y, Z, R, RJ, IERR)

B.6.a Argument Definitions

X, Y, Z, R [in] Arguments of the elliptic integral. Require X ≥ 0, Y ≥ 0, Z ≥ 0, at most one of X, Y or Z equal zero, R ≠ 0. See Section E.

RJ [out] The computed value of R_J(X, Y, Z, R).

IERR [out] Status indicator:

0 = no errors

1 = X < 0.0 or Y < 0.0 or Z < 0.0 or R = 0.0.

2 = X + Y or X + Z or Y + Z or |R| too small (See Section E).

3 = X or Y or Z or |R| too large (See Section E).

B.7 Modifications for Double Precision

For double precision usage, change the REAL type statements to DOUBLE PRECISION and change the subprogram names SELEFI, SELPII, SRCVAL, SRDVAL, SRFVAL and SRJVAL to DELEFI, DELPII, DRCVAL, DRDVAL, DRFVAL and DRJVAL, respectively.

C. Examples and Remarks

C.1 Related Functions

Logarithms, inverse circular functions and inverse hyperbolic functions can be expressed in terms of R_C, see [9, pp. 163, 186]:

$$\begin{array}{ll}
(\ln x)/(x-1) = R_C((\tfrac{1}{2}+\tfrac{1}{2}x)^2, x), & x > 0; \\
(\sin^{-1} x)/x = R_C(1-x^2, 1), & -1 \le x \le 1; \\
(\sinh^{-1} x)/x = R_C(1+x^2, 1), & -\infty < x < \infty; \\
(\cos^{-1} x)/(1-x^2)^{\frac{1}{2}} = R_C(x^2, 1), & 0 \le x \le 1; \\
(\cosh^{-1} x)/(x^2-1)^{\frac{1}{2}} = R_C(x^2, 1), & x \ge 1; \\
(\tan^{-1} x)/x = R_C(1, 1+x^2), & -\infty < x < \infty; \\
(\tanh^{-1} x)/x = R_C(1, 1-x^2), & -1 < x < 1; \\
\cot^{-1} x = R_C(x^2, x^2+1), & 0 \le x < \infty; \\
\coth^{-1} x = R_C(x^2, x^2-1), & x > 1.
\end{array}$$

The first seven of these allow computing nearly indeterminate forms with more accuracy than would be possible using the naïve formulation.

Heuman's lambda function [3] is a variant of Legendre's third integral:

$$\begin{aligned}
&\frac{\left(1-\cos^2\alpha\sin^2\beta\right)^{1/2}}{\cos^2\alpha\sin\beta\cos\beta}\Lambda(\alpha,\beta,\varphi) \\
&= \sin\varphi\, R_F(\cos^2\varphi, 1-\sin^2\alpha\sin^2\varphi, 1) \\
&\quad + \frac{\sin^2\alpha\sin^3\varphi}{3\left(1-\cos^2\alpha\sin^2\beta\right)} \times R_J\Big(\cos^2\varphi, \\
&\quad 1-\sin^2\alpha\sin^2\varphi, 1, 1-\frac{\sin^2\alpha\sin^2\varphi}{1-\cos^2\alpha\sin^2\beta}\Big) \qquad (11)
\end{aligned}$$

$$\begin{aligned}
&\frac{\pi}{2}\Lambda_0(\alpha,\beta) = \Lambda(\alpha,\beta,\pi/2) \\
&= \sin\beta\left[R_F(0,\cos^2\alpha,1) - \frac{1}{3}\sin^2\alpha R_D(0,\cos^2\alpha,1)\right] \\
&\times R_F\left(\cos^2\beta, 1-\cos^2\alpha\sin^2\beta, 1\right) - \frac{1}{3}\cos^2\alpha\sin^3\beta \\
&\times R_F(0,\cos^2\alpha,1)\times R_D(\cos^2\beta, 1-\cos^2\alpha\sin^2\beta, 1) \qquad (12)
\end{aligned}$$

The variants of Legendre's integrals used by Bulirsch in [4] and [5] are

$$el1(x,k_c) = xR_F\left(1, 1+k_c^2x^2, 1+x^2\right), \qquad (13)$$

$$\begin{aligned}
el2(x,k_c,a,b) &= axR_F\left(1, 1+k_c^2x^2, 1+x^2\right) \\
&+ \frac{1}{3}(b-a)x^3R_D\left(1, 1+k_c^2x^2, 1+x^2\right) \qquad (14)
\end{aligned}$$

$$\begin{aligned}
&ele3(x,k_c,p) = xR_F\left(1, 1+k_c^2x^2, 1+x^2\right) \\
&+ \frac{1}{3}(1-p)x^3R_J\left(1, 1+k_c^2x^2, 1+x^2, 1+px^2\right) \qquad (15)
\end{aligned}$$

$$cel(k_c,p,a,b) = aR_F(0,k_c^2,1) = \frac{1}{3}(b-pa)R_J\left(0,k_c^2,1,p\right) \qquad (16)$$

C.2 Which Procedure Should Be Used?

Several factors influence the choice of procedure. If one needs to write a simple program and use it once, one should probably choose the procedure that evaluates the functions in the form most similar to the way the problem is posed. If one needs to write a program that will have substantial use, one should usually prefer SELEFI to SRDVAL and SRFVAL, as the former is up to 30 times faster than the latter two. An exception to this rule occurs if one needs to compute $R_D(a,b,c)$ with $c < \max(a,b)$, in which case the parameters for SELEFI will be out of range. If accuracy is an issue but speed is not, one may prefer SRDVAL and SRFVAL to SELEFI, at least for computing $\mathrm{F}(\varphi,k)$. (See testing in Section D below).

SRCVAL is somewhat slower, on an IBM PC/AT with a numeric data processor, than using the equivalent Fortran intrinsic functions. This is no surprise, as most of the intrinsic functions are implemented by hardware. But the inverse hyperbolic functions are not. SRCVAL is roughly the same speed as the procedures in Chapter 2.1. As mentioned above, it may be advantageous to use SRCVAL to compute nearly indeterminate forms.

SELPII is implemented by using SRJVAL and SRFVAL (see Eqs. (9) and (10) above). Thus, there is no special advantage in speed or accuracy to one or the other. The sole criterion is how closely the forms of the functions evaluated directly by the procedures match the forms of the functions the user needs to evaluate.

D. Functional Description

D.1 Properties of the Functions

The first form given in Eqs. (2)–(4) is the Jacobi or algebraic form. When expressed in this form Eq. (2) is finite for all real and complex y, including ∞, has a simple pole of order 1 for $y = \infty$, and is logarithmically infinite for $y = 1/\alpha^2$.

D.2 Method of Computation

The procedure SELEFI is based upon a procedure ELLPI developed by Allan V. Hershey and modified by Alfred H. Morris, described in [6]. The procedure uses series expansions due to DiDonato and Hershey, described in [7]. The procedure SELPII is based upon a procedure EPI developed by Alfred H. Morris, described in [5]. It computes $\Pi(\varphi, k^2, \alpha^2)$ using Eqs. (9) and (10), as computed by SRFVAL and SRJVAL. The procedures SRCVAL, SRDVAL, SRFVAL and SRJVAL are based on procedures developed by B. C. Carlson and Elaine M. Notis, described in [8] and [9]. All of the referenced procedures were revised to be consistent with low level modules and naming conventions of MATH77.

D.3 Testing

The single precision programs for $\mathrm{E}(\varphi, k)$, $\mathrm{F}(\varphi, k)$, $R_D(a, b, c)$ and $R_F(a, b, c)$ were tested on an IBM PC/AT (using IEEE arithmetic) by comparison to double precision results, as described below. The relative precision of IEEE single precision arithmetic is $\rho = 2^{-23} \approx 0.119 \times 10^{-6}$.

The accuracy of procedure SELEFI was assessed by comparing its results to double precision results obtained by applying Eqs. (8) and (9), with $R_D(a, b, c)$ and $R_F(a, b, c)$ evaluated by DRDVAL and DRFVAL, respectively. The accuracy of procedures SRDVAL and SRFVAL was assessed by comparing their results to double precision results obtained by applying Eqs. (8) and (9), with $\mathrm{E}(\varphi, k)$ and $\mathrm{F}(\varphi, k)$ evaluated by DELEFI.

To test SELEFI, the rectangular region $0 \leq \varphi \leq \pi/2 \times 0 \leq k \leq 1$ of the $\varphi \times k$ plane was divided into 2000 regions, and a point was randomly selected in each region. To test SRDVAL and SRFVAL, the argument c was set to 1.0, the rectangular region $0 \leq a < 1 \times 0 \leq b < 1$ of the $a \times b$ plane was divided into 2000 regions, and a point was randomly selected in each region. The maximum relative and absolute errors are summarized in the following table.

Function	Max. Rel. Error	Max. Abs. Error
$\mathrm{E}(\varphi, k)$	$.82\rho$	$.98\rho$
$\mathrm{F}(\varphi, k)$	5.24ρ	15.91ρ
$R_D(a, b, 1)$	1.20ρ	3.01ρ
$R_F(a, b, 1)$	1.35ρ	2.55ρ

Errors in $\mathrm{F}(\varphi, k)$ increase as the arguments approach the infinite singularity at $\varphi = \pi/2$ and $k = 1$.

References

1. Milton Abramowitz and Irene A. Stegun, **Handbook of Mathematical Functions**, *Applied Mathematics Series 55*, National Bureau of Standards (1966) Chapter 17, 587–626.
2. Paul F. Byrd and Morris D. Friedman, **Handbook of Elliptic Integrals for Engineers and Scientists**, Springer Verlag, Berlin (1971).
3. H. Kuki, *Tables of complete elliptic integrals*, **J. Math. and Physics 20** (1941) 127–206.
4. Roland Bulirsch, *Numerical calculation of elliptic integrals and elliptic functions*, **Numerische Mathematik 7** (1965) 78–90.
5. Roland Bulirsch, *Numerical calculation of elliptic integrals and elliptic functions*, **Numerische Mathematik 13** (1969) 305–315.
6. Alfred. H. Morris, Jr., **NSWC Library of Mathematics Subroutines**. Technical Report NSWCDD/TR-92/425, Naval Surface Warfare Center, Dahlgren, VA 22448-5000 USA (Jan. 1993) 107–110.
7. Armido R. DiDonato and Allan V. Hershey, *New formulas for computing incomplete elliptic integrals of the first and second kind*, **J. ACM 6** (1959) 512–526.
8. B. C. Carlson, *Computing elliptic integrals by duplication*, **Numerische Mathematik 33** (1979) 1–16.
9. B. C. Carlson and Elaine M. Notis, *Algorithm 577: Algorithms for incomplete elliptic integrals [S21]*, **ACM Trans. on Math. Software 7**, 3 (Sept. 1981) 398–403.
10. B. C. Carlson, **Special Functions of Applied Mathematics**, Academic Press, New York (1977).

E. Error Procedures and Restrictions

The procedure SELEFI requires $|\varphi| \leq \pi/2$, and $|k| \leq 1$.

Procedure SELPII computes $\Pi(\varphi, k^2, \alpha^2)$ from $R_J(a, b, c, r)$ and $R_F(a, b, c)$ using Eqs. (9) and (10). The initial values for the arguments are $a = \cos^2 \varphi$, $b = 1 - k^2 \sin^2 \varphi$, $r = 1 - \alpha^2 \sin^2 \varphi$, and $c = \max(a, b, r)$. Then a, b and r are replaced by $a \times c$, $b \times c$ and $r \times c$, respectively. SELPII requires $|\varphi| \leq \pi/2$. Restrictions on k^2 and α^2 are enforced indirectly by restrictions on a, b and c imposed by SRFVAL and SRJVAL, described below.

The ranges for φ and k^2 can be extended using formulae 113.01, 113.02, 114.01, 115.01, 115.02, 160.02, 161.02 and 162.02 from [2], or formulae 17.4.1 through 17.4.18 from [1].

Denote the largest representable magnitude by Ω, and the smallest nonzero representable magnitude by ω. General restrictions on the arguments to procedures SRCVAL, SRDVAL, SRFVAL and SRJVAL were described above in Section B.

SRCVAL requires $\mathrm{X} + |\mathrm{Y}| \geq 5\omega,\ \mathrm{X} \leq \Omega/5,\ |\mathrm{Y}| \leq \Omega/5$, and, if $\mathrm{Y} < -2.236/\sqrt{\omega}$ it requires $\mathrm{X} \geq (\omega\Omega)^2/25$.

Denote the machine round-off level by ρ, that is, ρ is the smallest positive number such that the representation of $1+\rho$ is different from 1. Let ε be the solution of the equation $\rho = 3\varepsilon^6(1-\varepsilon)^{-3/2}$, $\Omega_D = 2\Omega^{-2/3}$ and $\omega_D = \varepsilon\omega^{-2/3}/10$. SRDVAL requires $\mathrm{X} + \mathrm{Y} \geq \omega_D$ and $\mathrm{Z} \geq \omega_D,\ \mathrm{X} \leq \Omega_D,\ \mathrm{Y} \leq \Omega_D$ and $\mathrm{Z} \leq \Omega_D$.

SRFVAL requires $\mathrm{X}+\mathrm{Y} \geq 5\omega,\ \mathrm{X}+\mathrm{Z} \geq 5\omega,\ \mathrm{Y}+\mathrm{Z} \geq 5\omega,\ \mathrm{X} \leq \Omega/5,\ \mathrm{Y} \leq \Omega/5$ and $\mathrm{Z} \leq \Omega/5$.

Let $\Omega_J = (\Omega/5)^{1/3}/5$ and $\omega_J = (5\omega)^{1/3}$. SRJVAL requires $\mathrm{X}+\mathrm{Y} \geq \omega_J,\ \mathrm{Y}+\mathrm{Z} \geq \omega_J,\ \mathrm{X}+\mathrm{Z} \geq \omega_J,\ |\mathrm{R}| \geq \omega_J,\ \mathrm{X} \leq \Omega_J,\ \mathrm{Y} \leq \Omega_J,\ \mathrm{Z} \leq \Omega_J$ and $|\mathrm{R}| \leq \Omega_J$.

The accessible ranges of the arguments may be extended beyond the ranges admissible in the procedures by using the homogeneity of the functions:

$$R_F(ka,kb,kc) = k^{-1/2}R_F(a,b,c), \quad \text{and}$$
$$R_J(ka,kb,kc,kr) = k^{-3/2}R_J(a,\ b,\ c,\ r).$$

If any of the restrictions above is violated, all procedures return an error indicator in the argument named IERR, and invoke the error message processor (see Chapter 19.2) with LEVEL = 0. The procedure ERMSET (see Chapter 19.2) may be used to affect the default error processing action.

F. Supporting Information

The source language for these subroutines is ANSI Fortran 77.

The procedures SELEFI and SELPII were written by W. V. Snyder in December 1990, based on earlier procedures described by Alfred H. Morris, Naval Surface Warfare Center, Dahlgren, VA in [5]. The procedures SRCVAL, SRDVAL, SRFVAL and SRJVAL were written by W. V. Snyder in December 1990, based on earlier procedures described by Carlson and Notis in [9].

Entry	Required Files
DELEFI	AMACH, DELEFI, DERM1, DERV1, DLNREL, ERFIN, ERMSG
DELPII	AMACH, DELPII, DERM1, DERV1, DRCVAL, DRFVAL, DRJVAL, ERFIN, ERMSG
DRCVAL	AMACH, DERM1, DERV1, DRCVAL, ERFIN, ERMSG
DRDVAL	AMACH, DERM1, DERV1, DRDVAL, ERFIN, ERMSG
DRFVAL	AMACH, DERM1, DERV1, DRFVAL, ERFIN, ERMSG
DRJVAL	AMACH, DERM1, DERV1, DRCVAL, DRFVAL, DRJVAL, ERFIN, ERMSG
SELEFI	AMACH, ERFIN, ERMSG, SELEFI, SERM1, SERV1, SLNREL
SELPII	AMACH, ERFIN, ERMSG, SELPII, SERM1, SERV1, SRCVAL, SRFVAL, SRJVAL
SRCVAL	AMACH, ERFIN, ERMSG, SERM1, SERV1, SRCVAL
SRDVAL	AMACH, ERFIN, ERMSG, SERM1, SERV1, SRDVAL
SRFVAL	AMACH, ERFIN, ERMSG, SERM1, SERV1, SRFVAL
SRJVAL	AMACH, ERFIN, ERMSG, SERM1, SERV1, SRCVAL, SRFVAL, SRJVAL

DRSELI

```
      program DRSELI
c>>1994-10-19 DRSELI Krogh  Changes to use M77CON
c>>1992-03-09 DRSELI WV Snyder Create separate single and double demos.
c>>1991-10-04 DRSELI WV Snyder JPL Original code.
c--S replaces "?": DR?ELI,?RCVAL,?ELEFI,?ELPII,?RDVAL,?RFVAL,?RJVAL
c
c     Demonstration driver for incomplete elliptic integral procedures.
c
      real            ALPHA2, E, F, K, K2, PHI, PI, R, RC, RD, RF, RJ
      real            SINPHI, T, U, X, Y, Z
      integer IERR
c
c     Compute arc sine x using ASIN and RC, for x = 0.5
c
      print *, 'Identities from write-up:'
      x = 0.5e0
      call srcval (1.0e0-x*x,1.0e0,rc,ierr)
      if (ierr.eq.0) then
         t = asin(x) - x * rc
         print '('' ASIN(0.5) - 0.5*RC(1-0.5**2,1)    ='',g15.8)', t
      else
         print '('' SRCVAL returns error signal '',i1)', ierr
      end if
c
c     Evaluate identities given by equations (8-10) in the write-up
c     with k**2 = 1/2, sin(phi)**2 = 1/4, alpha**2 = 1/2, c = 1.
c     From this, we have a = 3/4, b = r = 7/8.
c
      alpha2 = 0.5e0
      k = sqrt(0.5e0)
      k2 = 0.5e0
      sinphi = 0.5e0
      phi = asin(sinphi)
      r = 0.875e0
      x = 0.75e0
      y = 0.875e0
      z = 1.0e0
      call selefi (phi,k,f,e,ierr)
      if (ierr.ne.0) then
         print '('' SELEFI returns error signal '',i1)', ierr
         go to 99
      end if
      call selpii (phi,k2,alpha2,pi,ierr)
      if (ierr.ne.0) then
         print '('' SELPII returns error signal '',i1)', ierr
         go to 99
      end if
      call srdval (x,y,z,rd,ierr)
      if (ierr.ne.0) then
         print '('' SRDVAL returns error signal '',i1)', ierr
         go to 99
      end if
      call srfval (x,y,z,rf,ierr)
      if (ierr.ne.0) then
         print '('' SRFVAL returns error signal '',i1)', ierr
         go to 99
```

```
      end if
      call srjval (x,y,z,r,rj,ierr)
      if (ierr.ne.0) then
         print '('' SRJVAL returns error signal '',i1)', ierr
         go to 99
      end if
      u = sqrt(z**3) * rd
      t = 3.0e0 / (k2 * sinphi**3) * (f-e)
      r = (u-t) / u
      print '('' Equation (8), (LHS - RHS)/LHS     ='',g15.8)', r
      u = sqrt(z) * rf
      t = f / sinphi
      r = (u-t) / u
      print '('' Equation (9), (LHS - RHS)/LHS     ='',g15.8)', r
      u = sqrt(z**3) * rj
      t = 3 / (alpha2 * sinphi**3) * (pi - f)
      r = (u-t) / u
      print '('' Equation (10), (LHS - RHS)/LHS    ='',g15.8)', r
c
99    stop
      end
```

ODSELI

```
 Identities from write-up:
 ASIN(0.5) - 0.5*RC(1-0.5**2,1)     = 0.59604645E-07
 Equation (8), (LHS - RHS)/LHS      =-0.18963517E-05
 Equation (9), (LHS - RHS)/LHS      =   0.0000000
 Equation (10), (LHS - RHS)/LHS     = 0.23314153E-05
```